

Approximate Credal Network Updating by Linear Programming with Applications to Decision Making[☆]

Alessandro Antonucci^{a,*}, Cassio P. de Campos^b, David Huber^a, Marco Zaffalon^a

^a*Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA),
Galleria 2, 6928 Manno (Lugano), Switzerland.*

^b*Queen's University Belfast, School of Electronics, Electrical Engineering and Computer Science,
Computer Science at Elmwood ECS1, Elmwood Avenue, BT9 6AZ, UK, Belfast, United Kingdom.*

Abstract

Credal nets are probabilistic graphical models which extend Bayesian nets to cope with sets of distributions. An algorithm for approximate credal network updating is presented. The problem in its general formulation is a multilinear optimization task, which can be linearized by an appropriate rule for fixing all the local models apart from those of a single variable. This simple idea can be iterated and quickly leads to accurate inferences. A transformation is also derived to reduce decision making in credal networks based on the maximality criterion to updating. The decision task is proved to have the same complexity of standard inference, being NP^{PP}-complete for general credal nets and NP-complete for polytrees. Similar results are derived for the E-admissibility criterion. Numerical experiments confirm a good performance of the method.

Keywords: Credal Networks, Bayesian Networks, Linear Programming, Decision Making, Maximality, E-admissibility

1. Introduction

Credal networks (Cozman, 2000) are a generalization of Bayesian networks (Pearl, 1988) based on the notion of credal sets. A credal set is a set of probability mass functions, it represents quite a general and expressive model of uncertainty. Other uncertainty models like belief functions (Shafer, 1976) or possibility measures (Dubois and Prade, 1988) can be regarded as (special classes of) credal sets. A Bayesian network can be turned into a credal network by replacing the local models, which are conditional probability mass functions, with conditional credal sets over the same variables. Exactly as a Bayesian network defines a joint probability mass function over its whole

[☆]This paper is an extended and revised version of material originally presented in (Antonucci et al., 2013) and (Antonucci and de Campos, 2011).

*Corresponding author.

Email addresses: alessandro@idsia.ch (Alessandro Antonucci), c.decampos@qub.ac.uk (Cassio P. de Campos), david@idsia.ch (David Huber), zaffalon@idsia.ch (Marco Zaffalon)

set of variables, a credal network defines a joint credal set, which is (the convex closure of) the set of all joint mass functions obtained from the Bayesian networks consistent with the local credal sets.

Inference in credal networks is intended as the computation of the lower and upper bounds of expectations and probabilities with respect to that joint credal set. This makes inference considerably harder than in the case of Bayesian networks. For instance, a marginalization task corresponds to a multilinear optimization problem, while updating is a multilinear-fractional task (de Campos and Cozman, 2004). This problem is known to be NP-hard even for singly connected networks (de Campos and Cozman, 2005), while the analogous inference in Bayesian networks can be performed in polynomial time (e.g., Koller and N. Friedman, 2009). Despite the hardness of the problem, some algorithms are known to perform reasonably well under certain conditions. Exact approaches have been proposed that implement some branch-and-bound method with local searches (da Rocha et al., 2003; de Campos and Cozman, 2005; Cano et al., 2007; de Campos and Cozman, 2007). Unfortunately they all suffer from serious efficiency issues unless the credal network is very simple. For instance, none of these methods can deal well with a binary node having four ternary parents, because this setting is already equivalent to $3^4 = 81$ free optimization variables to be chosen, meaning a space of 2^{81} possible solutions just locally on this node! On the other hand, approximate methods either are fast and provide no accuracy guarantee (Cano et al., 2007; da Rocha et al., 2003; Antonucci et al., 2010) or provide theoretical guarantees but are as slow as exact methods (Mauá et al., 2012a). Moreover, all these approximate methods are only capable of treating credal networks whose credal sets are specified by enumerating the extreme points, while a constraint-based specification still lacks any practical algorithm. An exception is the exact algorithm for trees by de Cooman et al. (2010), which considers the independence concept of epistemic irrelevance, while the more popular notion of strong independence is considered here (see Sect. 4.5 for a more detailed discussion on this point).

In this paper we present a fast approximate algorithm, called A-LP, for inferences in credal networks based on solving a sequence of linear programming problems. It uses a constraint-based specification, which allows us to deal with domains where the local credal sets are given by their linear constraints. It does not suffer from large credal sets because the optimization is done by compact linear problems, and the complexity with respect to the topology is the same as in the Bayesian network case. To the best of our knowledge, this is the first method for general credal networks to truly run the inference with a constraint-based specification. We describe the method and some heuristic ideas to improve its accuracy. Unlike similar ideas already proposed by da Rocha et al. (2003), our approach does not require an explicit enumeration of the extreme points of the credal sets and should be therefore used when the number of extreme points in the local credal sets is exponentially large (e.g., variables with many states and/or parents, credal sets defined by probability intervals). We also discuss decision making in credal networks based on the maximality criterion (Walley, 1991). We evaluate the computational complexity of this decision task, and show how to reduce the problem to updating by simple graphical transformations. An approximate approach to the decision problem can be therefore achieved by means of the proposed algorithm. Similar results are obtained for the E-admissibility criterion (Levi, 1980).

The paper is organized as follows. Sects. 2 and 3 review the basic notation and definitions of Bayesian and credal networks. The proposed procedure is presented in Sect. 4 and extended to decision making in Sect. 5. Numerical experiments in Sect. 6 show that the proposed method compares favorably against other available methods in the literature. Conclusions and outlooks are in Sect. 7, while the proofs are in the appendix.

2. Bayesian networks

Consider a set of variables $\mathbf{X} := (X_0, X_1, \dots, X_n)$ in one-to-one correspondence with the nodes of an acyclic directed graph \mathcal{G} . For each $i = 0, \dots, n$, the joint variable $\Pi_i \subset \mathbf{X}$ denotes the parents of X_i according to \mathcal{G} . All these variables are categorical: X_i takes its values on the finite set Ω_{X_i} and so does Π_i in $\Omega_{\Pi_i} := \times_{X_j \in \Pi_i} \Omega_{X_j}$, for each $i = 0, \dots, n$ (symbol \times denotes Cartesian set product). The graph \mathcal{G} represents stochastic independence relations by means of a Markov condition: any variable is conditionally independent of its non-descendant non-parents given its parents (e.g., Koller and N. Friedman, 2009). The specification of a conditional probability mass function $P(X_i|\pi_i)$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \dots, n$, induces through the graph, for each $\mathbf{x} \in \Omega_{\mathbf{X}} := \times_{i=0}^n \Omega_{X_i}$, the factorization:

$$P(\mathbf{x}) := \prod_{i=0}^n P(x_i|\pi_i), \quad (1)$$

where the values of x_i and π_i are those consistent with \mathbf{x} . A specification of the conditional probability mass functions $\{P(X_i|\pi_i)\}_{i=0, \dots, n}^{\pi_i \in \Omega_{\Pi_i}}$, together with Eq. (1) (or the graph) is called a Bayesian network.

In particular, the mass functions associated with X_i , i.e., $\{P(X_i|\pi_i)\}_{\pi_i \in \Omega_{\Pi_i}}$ are called the *local models* of X_i , for each $i = 0, \dots, n$. Inference in Bayesian networks is based on the joint probability mass function in Eq. (1). Marginals, for instance, are obtained by summing out other variables from the joint, e.g., the marginalization of X_0 corresponds to the computation, for each $x_0 \in \Omega_{X_0}$, of:

$$P(x_0) = \sum_{x_1, \dots, x_n} \prod_{i=0}^n P(x_i|\pi_i), \quad (2)$$

where $\sum_{\mathbf{x}}$ is a shortcut notation for $\sum_{\mathbf{x} \in \Omega_{\mathbf{X}}}$. Alternatively, the marginal in Eq. (2) can also be expressed as a linear combination of the local probabilities associated with an arbitrary $X_j \in \mathbf{X}$, i.e.,

$$P(x_0) = \sum_{x_j, \pi_j} [P(x_0|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j), \quad (3)$$

where the probabilities $P(x_j|\pi_j)$ are already available in the Bayesian network specification, the unconditional probabilities $P(\pi_j)$ are computed as in Eq. (2), and for the conditional ones $P(x_0|x_j, \pi_j) = P(x_0, x_j, \pi_j)/P(x_j, \pi_j)$, assuming $P(x_j, \pi_j) > 0$.

If $X_j = X_0$, Eq. (3) becomes $P(x_0) = \sum_{\pi_0} P(\pi_0) \cdot P(x_0|\pi_0)$, and, if $X_0 \in \Pi_j$, then it becomes $P(x_0) = \sum_{x_j, \pi'_j} P(x_0, \pi'_j) P(x_j|x_0, \pi'_j)$, with $\Pi'_j := \Pi_j \setminus \{X_0\}$.

The values of $P(\pi_j)$ and $P(x_0|x_j, \pi_j)$, for each $x_j \in \Omega_{X_j}$ and $\pi_j \in \Omega_{\Pi_j}$, are not affected by those of the local models of X_j . To see that, note that when computing a marginal, removing the descendants and their local models does not affect the probability. As X_j is a child of all the variables in Π_j , the computation of $P(\pi_j)$ is not affected by the values of $\{P(X_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_j}}$. Similarly, when computing a conditional probability, arcs leaving the variables after the conditioning bar can be removed: thus, in the case of $P(x_0|x_j, \pi_j)$, we can disconnect X_j from the rest of the network, thus making its local model irrelevant for the particular calculation. This remark, together with Eq. (3) will be exploited by the A-LP approximate algorithm presented later.

3. Credal sets and credal networks

3.1. Credal sets

The Bayesian theory of subjective probability has been extended by more general uncertainty theories in order to model situations of highly incomplete or conflicting information. Among others, the theory of imprecise probability (Walley, 1991) adopts credal sets, which are closed and convex sets of probability mass functions, as a more general model of uncertainty about the state of a categorical variable.

In particular, here we focus on finitely generated credal sets, which are specified by a finite number of linear constraints on the probabilities (e.g., see Fig. 1). A credal set over X is denoted here as $K(X)$, while its extreme points (i.e., points of the set which cannot be expressed as a convex combination of other ones) are denoted as $\text{ext}[K(X)]$. As we cope with finitely generated credal sets, the number of extreme points is finite.

Expectations. Given a single probability mass function $P(X)$, the expectation of a real-valued function, or *gamble* in the behavioural jargon, $g : \Omega_X \rightarrow \mathbb{R}$ is $E(g) := \sum_x P(x) \cdot g(x)$. If $K(X)$ is given instead, only the bounds with respect to the mass functions in the credal set can be evaluated, i.e., for the lower bound:

$$\underline{E}(g) := \min_{P(X) \in K(X)} \sum_{x \in \Omega_X} P(x) \cdot g(x), \quad (4)$$

and similarly with a maximum replacing the minimum for the upper bound $\overline{E}(g)$. As $K(X)$ is defined by linear constraints, the optimization in Eq. (4) is a linear programming task. It is well known that the optimum is achieved in $\text{ext}[K(X)]$. We will exploit this property further on.

Decision making. Given a single probability mass function $P(X)$, the optimal option under zero-one loss, i.e., the most probable state, is $x^* := \arg \max_{x \in \Omega_X} P(x)$. When coping with credal sets, different mass functions in $K(X)$ might assign the highest probability to different options, and there is not a single way to extend the decision criterion adopted in the Bayesian framework. The so-called Γ -maximin criterion takes the options maximizing the lower probability, i.e., those such that $x^* := \arg \max_{x \in \Omega_X} \underline{P}(x)$, where $\underline{P}(x)$ is the lower expectation of the indicator associated

with x . Γ -maximax does the same with the upper probabilities replacing the lower ones. Alternatively, the overlap among the ranges of the probabilities can be considered: states whose upper probabilities is smaller than the lower of some other state are discarded. This is called interval dominance, and leads to the following set $\Omega_X^* \subseteq \Omega_X$ of optimal options:¹

$$\Omega_X^* := \{x' \in \Omega_X \mid \nexists x'' \in \Omega_X : \underline{P}(x'') > \overline{P}(x')\}. \quad (5)$$

According to the maximality criterion, a state is discarded if, for each element of the credal set, there is another state with higher probability, i.e.,

$$\Omega_X^* := \{x' \in \Omega_X \mid \nexists x'' \in \Omega_X : P(x'') > P(x'), \forall P(X) \in K(X)\}. \quad (6)$$

Another approach is E-admissibility, where only options which are optimal for at least a probability mass function of the credal set are adopted, i.e.,

$$\Omega_X^* := \left\{x' \in \Omega_X \mid \exists P(X) \in K(X) : P(x') \geq \max_{x'' \in \Omega_X} P(x'')\right\}. \quad (7)$$

Let us use a superscript to distinguish between the sets of optimal options based on the different criteria. The following chain of inclusions holds:

$$\Omega_X^{\Gamma\text{-maximax}} \subseteq \Omega_X^{\text{E-admissibility}} \subseteq \Omega_X^{\text{maximality}} \subseteq \Omega_X^{\text{interval dominance}}, \quad (8)$$

together with:

$$\Omega_X^{\Gamma\text{-maximin}} \subseteq \Omega_X^{\text{maximality}}. \quad (9)$$

The proofs can be found in (Troffaes, 2007). Despite a substantial lack of unanimity about the best way to take decisions based on credal sets, maximality and E-admissibility seem a reasonable compromise between the necessary caution when taking decisions and the need of avoiding too a high number of possible options.

3.2. Credal networks

Credal sets can be used to extend Bayesian networks to imprecise probabilities. To do that, in the definition of a Bayesian network, every conditional probability mass function $P(X_i|\pi_i)$ is replaced by a conditional credal set $K(X_i|\pi_i)$. A specification of the conditional credal sets $\{K(X_i|\pi_i)\}_{i=0,\dots,n}^{\pi_i \in \Omega_{\Pi_i}}$ together with the graph \mathcal{G} is called a credal network. Under this generalized setting, Eq. (1) can be used to obtain different joint probability mass functions. Let us consider all the possible combinations of the extreme points in the local models, and then take the convex hull (denoted as CH), i.e., build the following joint credal set:

$$K(\mathbf{X}) := \text{CH} \left\{ P(\mathbf{X}) \mid P(\mathbf{x}) := \prod_{i=0}^n P(x_i|\pi_i), \begin{array}{l} \forall \mathbf{x} \in \times_{i=0}^n \Omega_{X_i}, \\ \forall P(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)] \\ \forall i = 0, \dots, n, \forall \pi_i \in \Omega_{\Pi_i} \end{array} \right\}. \quad (10)$$

¹The set in Eq. (5) can be equivalently obtained as $\{x' \in \Omega_X \mid \overline{P}(x') \geq \max_{x'' \in \Omega_X} \underline{P}(x'')\}$. This makes the identification time linear instead of quadratic in the number of options.

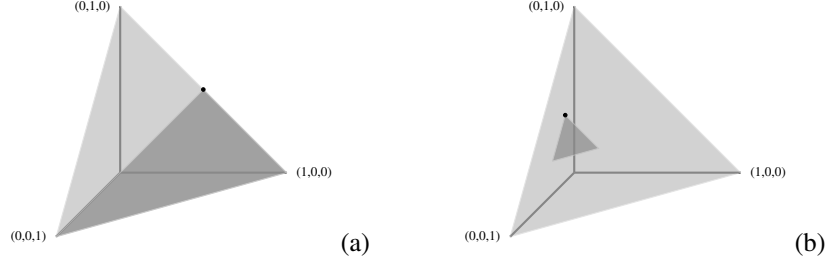


Figure 1: Credal sets over a ternary variable X (i.e., $\Omega_X = \{x', x'', x'''\}$). The representation is in a three-dimensional space with coordinates $[P(x'), P(x''), P(x''')]$. The dark grey polytopes represent respectively: (a) the credal set defined by constraint $P(x') \geq P(x'')$; (b) a credal set whose extreme points are $\{[.1, .3, .6]^T, [.3, .3, .4]^T, [.1, .5, .4]^T\}$. Note that $\bar{P}(x'') = .5$ for both credal sets, which corresponds to the black extreme point $[\cdot 5, \cdot 5, 0]^T$ of the first credal set and $[\cdot 1, \cdot 5, \cdot 4]^T$ for the second.

The credal set in Eq. (10) is called the strong extension of the credal network (Cozman, 2000). The name comes from strong independence, an independence concept for credal sets requiring stochastic independence on the extreme points. Each extreme point of the strong extension factorizes as in Eq. (1) and hence satisfies the stochastic independence relations induced by \mathcal{G} (see, e.g., Prop. 1 in Antonucci and Zaffalon, 2008). The strong extension of a credal network represents a generalization, based on credal sets, of the joint probability mass function associated with a Bayesian network as in Eq. (1). Inference in credal networks is based on the strong extension. For instance, the lower bound of the marginal probability in Eq. (2) is:

$$\underline{P}(x_0) := \min_{P(\mathbf{X}) \in K(\mathbf{X})} P(x_0) = \min_{\substack{P(X_i | \pi_i) \in K(X_i | \pi_i) \\ \pi_i \in \Omega_{\Pi_i}, i=0, \dots, n}} \sum_{x_1, x_2, \dots, x_n} \prod_{i=0}^n P(x_i | \pi_i), \quad (11)$$

and similarly for $\bar{P}(x_0)$. Eq. (11) represents a non-linear optimization problem, with a multilinear objective function over a feasible region defined by linear constraints on the optimization variables. In the next section we present an approximate algorithm to solve such a problem.

4. The A-LP algorithm

4.1. Solving the linear program

The idea. The A-LP algorithm presented here is based on the ideas of Lukatskii and Shapot (2000) to approximate the solution of multilinear problems. In essence, a multilinear optimization can be converted into a linear one if we fix all but one optimization variable in each of its multilinear terms. In Lukatskii and Shapot's terminology, there is a partition of the optimization variables such that, fixing the optimization variables in every set of the partition apart from one, the multilinear problem becomes linear. By iterating over the index defining the set to remain free, one can approximate the solution of the multilinear problem with a sequence of linear ones.

The linear program. Following the above mentioned ideas, we reduce the multilinear optimization to a linear one by shrinking all the local credal sets to singletons, namely extreme points, apart from those of a variable $X_j \in \mathbf{X}$, which we call the *free* variable. Given X_j , we therefore pick an extreme point $\tilde{P}(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)]$, for each $i = 0, \dots, n$ such that $i \neq j$ and each $\pi_i \in \Omega_{\Pi_i}$. These are additional constraints to the optimization problem in Eq. (11), which becomes:

$$\begin{aligned} \underline{P}'(x_0) &:= \min_{\substack{P(X_j|\pi_j) \in K(X_j|\pi_j) \\ \pi_j \in \Omega_{\Pi_j}}} \sum_{x_1, x_2, \dots, x_n} \left[\prod_{i=0, i \neq j}^n \tilde{P}(x_i|\pi_i) \right] \cdot P(x_j|\pi_j) = \\ &= \min_{\substack{P(X_j|\pi_j) \in K(X_j|\pi_j) \\ \pi_j \in \Omega_{\Pi_j}}} \sum_{x_j, \pi_j} \left[\tilde{P}(x_0|x_j, \pi_j) \cdot \tilde{P}(\pi_j) \right] \cdot P(x_j|\pi_j), \end{aligned} \quad (12)$$

where the last derivation is based on Eq. (3), and probabilities $\tilde{P}(x_0|x_j, \pi_j)$ and $\tilde{P}(\pi_j)$ are denoted by a tilde as they are computed from the joint of a Bayesian network with local models $\{\tilde{P}(X_i|\pi_i)\}_{\pi_i \in \Omega_{\Pi_i}}$ for each $i \neq j$, while the local models of X_j can be arbitrarily specified as already discussed at the end of Sect. 2.

Let us comment on some important facts about Eq. (12). Firstly, being the solution of an optimization with additional constraints with respect to Eq. (11) (see the second term in the equation), we have $\underline{P}'(x_0) \geq \underline{P}(x_0)$. Secondly, it is clear from the third term of Eq. (12) that the computation of $\underline{P}'(x_0)$ is a linear program whose optimization variables are the local probabilities of X_j , i.e., $\{P(x_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_j}, x_j \in \Omega_{X_j}}$. Lastly, as the solution of a linear program lies on an extreme point of the feasible region, there is a $P^*(X_j|\pi_j) \in \text{ext}[K(X_j|\pi_j)]$, for each $\pi_j \in \Omega_{\Pi_j}$ such that:

$$\underline{P}'(x_0) = \sum_{x_1, x_2, \dots, x_n} \left[\prod_{i=0, i \neq j}^n \tilde{P}(x_i|\pi_i) \right] \cdot P^*(x_j|\pi_j). \quad (13)$$

4.2. Searching for the optimum

The procedure detailed in the previous section provides an upper approximation of the lower probability $\underline{P}(x_0)$. A credal network whose local credal sets are singletons apart from those associated with a single variable can be called an *almost-Bayesian network*. The proposed optimization consists therefore in taking an almost-Bayesian network consistent with the original credal network (i.e., its strong extension is included in that of the original credal network) and exploiting the fact that marginalization of almost-Bayesian networks is a linear problem. By solving this linear problem, we obtain: (i) an upper approximation of the lower probability; (ii) a specification of the extreme points of the credal sets associated with the only ‘non-Bayesian’ variable in the almost-Bayesian network leading to the optimum. The extreme points in (ii) can be an assignment for the corresponding credal sets, and another variable can be freed, leading to a new linear program.

The above procedure can be arbitrarily iterated. This implements a *local search* approach to the optimization task in Eq. (11). The neighborhood concept here is less

sparse than that considered by da Rocha et al. (2003). Instead of considering all the extreme configurations of the models, only the one leading to the best solution for each variable is regarded as a neighbor. This makes unnecessary an explicit enumeration of the extreme points. In the rest of this section we suggest a possible architecture for the iterative process based on standard strategies for discrete optimization (Glover and Kochenberger, 2003).

Initialization. Before any iteration, the optimization in Eq. (12) requires an almost-Bayesian network consistent with the credal network. The local credal sets are defined by linear constraints and enumerating their extreme points can be unfeasible. We therefore generate an extreme point of each credal set as the solution of a linear program whose feasible region is the credal set itself and with a random objective function.

Improving the solution. The solution in Eq. (13) of the linear program in Eq. (12) provides an approximate solution of Eq. (11). As already noticed, to iterate the procedure another ‘free’ variable can be picked and the optimal solution $\{P^*(X_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_j}}$ of the previous problem can be used for a new initialization. The effect is shown here.

Proposition 1. Let $\{\tilde{P}(X_i|\pi_i)\}_{i=0,1,\dots,n}^{\pi_i \in \Omega_{\Pi_i}}$ be a Bayesian network specification consistent with a credal network specification $\{K(X_i|\pi_i)\}_{i=0,1,\dots,n}^{\pi_i \in \Omega_{\Pi_i}}$. As in Eq. (2), let $\tilde{P}(x_0) := \sum_{x_1, \dots, x_n} \prod_{i=0}^n \tilde{P}(x_i|\pi_i)$ and, as in Eq. (12),

$$\tilde{P}'(x_0) := \min_{\substack{P(X_j|\pi_j) \in K(X_j|\pi_j) \\ \pi_j \in \Omega_{\Pi_j}}} \sum_{x_1, \dots, x_n} \left[\prod_{\substack{i=0 \\ i \neq j}}^n \tilde{P}(x_i|\pi_i) \right] P(x_j|\pi_j). \quad (14)$$

Then $\tilde{P}'(x_0) \leq \tilde{P}(x_0)$.

The proof of this proposition is in Appendix A, together with those of all the other technical results presented in this paper. As a corollary of Prop. 1, it follows that each iteration of our procedure produces a better (or equal) approximation.

Iteration architecture. Before any iteration, we only have to choose the variable to be freed. As any choice can only improve the quality of the approximation, randomly picking a free variable at each iteration with no particular strategies could be an option. Yet, to decide whether or not variable X_j should be the free variable, we can better compare the improved solution $\tilde{P}'_j(x_0)$ obtained by freeing X_j with the previous candidate solution $\tilde{P}'(x_0)$. A greedy strategy consists in picking the variable leading to the best improvement, i.e., X_{j^*} where $j^* := \arg \min_{j=0, \dots, n} \tilde{P}'_j(x_0)$. A lazy alternative is to pick as free variable the first (according to some random ordering) improving the approximation, i.e., such that $\tilde{P}'_j(x_0) < \tilde{P}'(x_0)$. As far as we know there is no evidence in the literature to support that one is always preferred to the other, so in the rest of this paper we employ the lazy approach.

Stopping rule. The above described iteration strategy needs a stopping criterion. The simplest thing to do is to keep iterating the algorithm unless it is not possible to further improve the approximation. According to Prop. 1, this means that the procedure stops when the next iteration returns the same approximation, no matter which one is the picked free variable. This can be a local (or global) optimum, but also a non-optimal saddle point. To avoid stops in saddle points, we keep iterating (for a while) even if no improvements cannot be achieved. The choice of the free variable cannot be anymore based on the improvement in the approximation, which is zero for all variables. Thus, we just select, again in a lazy way, the first free variable leading to a different specification of the extreme points for the local models (even if the same approximation is achieved). The algorithm stops when, no matter the variable we free, the same solution and the same extreme points are returned. Regardless of that, we stop the algorithm after a maximum number of iterations for which the approximation cannot be improved.

Random restarts. To achieve robust results with respect to the initialization of the extreme points, we repeat the search process a number of times with different initializations. As the objective functions used to initialize the extreme points of the local credal sets are randomly specified, multiple runs produce different initializations. The above described iteration part is therefore executed separately for each initialization and the best (i.e., for lower probabilities, smallest) result is eventually returned.

Pseudocode. Alg. 1 contains the pseudocode of the overall algorithm with both the iteration strategy and the random restart procedure. We call the algorithm A-LP (*approximation with linear programming*). We randomly restart the search t times (line 3). Variables p and pp are, respectively, the approximation associated with the current restart and the minimum over all the restarts returned in output (line 39). These variables are initialized to 1.0 as their values can only decrease when iterating. The initialization of the extreme points of the local credal sets (line 7) is based on the linear programs with randomly specified objective function. A candidate free variable is randomly picked from a replica \mathbf{X}' of \mathbf{X} (line 10). If freeing the variable improves the accuracy (line 12), the variable is actually freed and the extreme points associated with the solution used in the next iteration (line 14). If this is not the case the actual variable is removed from the list of candidates \mathbf{X}' (line 17). If no candidate remains available (line 19), we pick from a second replica \mathbf{X}'' of \mathbf{X} (line 20) by looking for a variable such that the linear program, although leading to the same approximation, returns a different specification of the extreme points (line 22). The iteration stops if even the second list of candidates remains empty (line 36), or if the counter b of consecutive iterations without improvements reaches the threshold s (line 32).

By changing the direction of the inequalities in lines 12, 24, and 40, and computing the maximum instead of the minimum in the linear programs (lines 11 and 21), a lower approximation $\bar{P}'(x_0)$ of the upper probability $\bar{P}(x_0)$ is obtained instead. Overall, A-LP finds an inner approximation of the interval $[P(x_0), \bar{P}(x_0)]$, i.e., an upper approximation of the lower probability as in Eq. (11) and a lower approximation of the corresponding upper probability. The extension of these ideas to the computation

Algorithm 1 The A-LP algorithm

[Parameters] s (maximum number of no-improve iterations) and t (number of restarts)

[Input] a credal network specification $\{K(X_i|\pi_i)\}_{i=0,\dots,n}^{\pi_i \in \Omega_{\Pi_i}}$

[Output] p an upper approximation of $\underline{P}(x_0)$

```
1:  $pp \leftarrow 1.0$ 
2:  $b \leftarrow 0$ 
3: for  $count \leftarrow 1, t$  do ▷ random restarts
4:    $p \leftarrow 1.0$ 
5:    $\mathbf{X}', \mathbf{X}'' \leftarrow \mathbf{X}$ 
6:    $iterate \leftarrow \text{TRUE}$ 
7:    $\tilde{P}(X_i|\pi_i) \leftarrow \text{randomly pick from ext}[K(X_i|\pi_i)] \forall i, \pi_i$  ▷ initialization
8:   while  $iterate$  do
9:     if  $\mathbf{X}' \neq \emptyset$  then
10:       $X_j \leftarrow \text{randomly pick from } \mathbf{X}'$ 
11:       $\underline{P}'(x_0), P^*(X_j|\pi_j) \leftarrow \text{LP with } X_j, \{P(X_i|\pi_i)\}_{i \neq j}$  ▷ Eqs. (11) (12)
12:      if  $\underline{P}'(x_0) < p$  then
13:         $p \leftarrow \underline{P}'(x_0)$ 
14:         $\tilde{P}(X_j|\pi_j) \leftarrow P^*(X_j|\pi_j) \forall \pi_j$ 
15:         $\mathbf{X}' \leftarrow \mathbf{X} \setminus \{X_j\}$ 
16:      else
17:         $\mathbf{X}' \leftarrow \mathbf{X}' \setminus \{X_j\}$ 
18:      end if
19:    else
20:       $X_j \leftarrow \text{randomly pick from } \mathbf{X}''$ 
21:       $\underline{P}'(x_0), P^*(X_j|\pi_j) \leftarrow \text{LP with } X_j, \{P(X_i|\pi_i)\}_{i \neq j}$  ▷ as in line 11
22:      if  $\tilde{P} \neq P^*$  then
23:         $\tilde{P}(X_j|\pi_j) \leftarrow P^*(X_j|\pi_j) \forall \pi_j$ 
24:        if  $\underline{P}'(x_0) < p$  then
25:           $p \leftarrow \underline{P}'(x_0)$ 
26:           $\mathbf{X}' \leftarrow \mathbf{X} \setminus \{X_j\}$ 
27:           $\mathbf{X}'' \leftarrow \mathbf{X}$ 
28:           $b \leftarrow 0$ 
29:        else
30:           $b++$ 
31:           $\mathbf{X}'' \leftarrow \mathbf{X} \setminus \{X_j\}$ 
32:          if  $b \equiv s$  then  $iterate \leftarrow \text{FALSE}$  ▷ too many no-improvements
33:          end if
34:        else
35:           $\mathbf{X}'' \leftarrow \mathbf{X}'' \setminus \{X_j\}$ 
36:          if  $\mathbf{X}'' \equiv \emptyset$  then  $iterate \leftarrow \text{FALSE}$  ▷ no way to continue
37:          end if
38:        end if
39:      end while
40:      if  $p < pp$  then  $pp \leftarrow p$  ▷ take the best
41:    end for
42:  return  $pp$ 
```

of posterior (instead of marginal) probabilities deserves a specific discussion that is presented in the next section.

4.3. Extension to the conditional case

While the version of the A-LP algorithm presented in the previous section is designed to approximate lower and upper marginal probabilities, a typical inference task in Bayesian and credal networks is the evaluation of the posterior beliefs about a variable given an observation of some other variables. This task is called *updating*. Without lack of generality, X_0 is still the variable of interest. The observed variables are $\mathbf{X}_E \subseteq \mathbf{X} \setminus \{X_0\}$ and $\mathbf{x}_E \in \Omega_{\mathbf{X}_E}$ is the outcome of the observation. For Bayesian networks, the task is to compute $P(x_0|\mathbf{x}_E) = P(x_0, \mathbf{x}_E)/P(\mathbf{x}_E)$, provided that $P(\mathbf{x}_E) > 0$. This generalizes to credal networks as follows:

$$\underline{P}(x_0|\mathbf{x}_E) = \min_{\substack{P(X_i|\pi_i) \in K(X_i|\pi_i) \\ \pi_i \in \Omega_{\Pi_i}, i=0, \dots, n}} \frac{\sum_{\mathbf{x}'} \prod_{i=0}^n P(x_i|\pi_i)}{\sum_{x_0, \mathbf{x}'} \prod_{i=0}^n P(x_i|\pi_i)}. \quad (15)$$

where $\mathbf{X}' := \mathbf{X} \setminus (\{X_0\} \cup \mathbf{X}_E)$, and a similar relation holds for $\bar{P}(x_0|\mathbf{x}_E)$. Thus, while the marginal computation in Eq. (11) is a multilinear task, the posterior corresponds to a multilinear-fractional problem. Unlike the case of Bayesian networks, credal networks updating cannot be reduced to the computation of (joint) marginals. Mauá et al. (2014) provide an analysis of the differences between marginal and conditional queries in credal networks from the point of view of computational complexity.

To apply A-LP to conditional queries like in Eq. (15), we manipulate the Bayesian expression as in the marginal case. Given a Bayesian network and a $X_j \in \mathbf{X}$, the analogous of Eq. (3) with (x_0, \mathbf{x}_E) instead of x_0 is:

$$P(x_0, \mathbf{x}_E) = \sum_{x_j, \pi_j} [P(x_0, \mathbf{x}_E|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j). \quad (16)$$

Summing over all the $x_0 \in \Omega_{X_0}$, we obtain:

$$P(\mathbf{x}_E) = \sum_{x_j, \pi_j} [P(\mathbf{x}_E|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j). \quad (17)$$

Thus, the posterior probability, i.e., the objective function in Eq. (15), is:

$$P(x_0|\mathbf{x}_E) = \frac{\sum_{x_j, \pi_j} [P(x_0, \mathbf{x}_E|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j)}{\sum_{x_j, \pi_j} [P(\mathbf{x}_E|x_j, \pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j)}. \quad (18)$$

It is therefore possible to shrink all the local credal sets associated with the non-free variables to single extreme points, say $\tilde{P}(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)]$ for each $i = 0, 1, \dots, n$, $i \neq j$ and $\pi_i \in \Omega_{\Pi_i}$, and leave unchanged those of the free variable X_j . This reduces the multilinear-fractional task in Eq. (15) to the following linear-fractional problem:

$$\underline{P}'(x_0|\mathbf{x}_E) := \min_{\substack{P(X_j|\pi_j) \in K(X_j|\pi_j) \\ \pi_j \in \Omega_{\Pi_j}}} \frac{\sum_{\mathbf{x}'} \left[\prod_{i=0, \dots, n}^{i \neq j} \tilde{P}(x_i|\pi_i) \right] \cdot P(x_j|\pi_j)}{\sum_{x_0, \mathbf{x}'} \left[\prod_{i=0, \dots, n}^{i \neq j} \tilde{P}(x_i|\pi_i) \right] \cdot P(x_j|\pi_j)}, \quad (19)$$

which, exploiting that $\tilde{P}(x_0, \mathbf{x}_E | x_j, \pi_j) = \tilde{P}(x_0 | \mathbf{x}_E, x_j, \pi_j) \cdot \tilde{P}(\mathbf{x}_E | x_j, \pi_j)$, and because of Eq. (18), rewrites as follows:

$$\min_{\substack{P(X_j | \pi_j) \in K(X_j | \pi_j) \\ \pi_j \in \Omega_{\Pi_j}}} \frac{\sum_{x_j, \pi_j} \left[\tilde{P}(x_0 | \mathbf{x}_E, x_j, \pi_j) \cdot \tilde{P}(\mathbf{x}_E | x_j, \pi_j) \cdot \tilde{P}(\pi_j) \right] \cdot P(x_j | \pi_j)}{\sum_{x_j, \pi_j} \left[\tilde{P}(\mathbf{x}_E | x_j, \pi_j) \cdot \tilde{P}(\pi_j) \right] \cdot P(x_j | \pi_j)}. \quad (20)$$

We have already shown in Sect. 2 that the computation of $\tilde{P}(\pi_j)$ in a Bayesian network with local models $\tilde{P}(X_i | \pi_i)$ for the non-free variables is not affected by the local models of the free variable X_j . The same holds for any probability conditional on (x_j, π_j) , like $\tilde{P}(x_0 | \mathbf{x}_E, x_j, \pi_j)$ and $\tilde{P}(\mathbf{x}_E | x_j, \pi_j)$. Overall, Eq. (20) corresponds to a linear-fractional program, whose feasible region is defined by the linear constraints defining the local models of X_j and such that the coefficients of the (linear-fractional) objective function can be computed by standard inferences in a Bayesian network with the same structure of the original credal network. The linear-fractional task can be eventually reduced to a linear one by the classical transformation of Charnes and Cooper (1962).²

Special cases. Let us discuss the special situations with respect to the choice of the free variable and show how these do not affect the applicability of our procedure.

- (i) The free variable is the query (i.e., $j = 0$). The term $\tilde{P}(x_0 | \mathbf{x}_E, x_j, \pi_j)$ in Eq. (20) is one if $x_j = x_0$ and zero otherwise. The sum over X_j in the numerator can be therefore removed and only the term associated with x_0 remains.
- (ii) The query is a parent of the free variable (i.e., $X_0 \in \Pi_j$). Let $\Pi'_j := \Pi_j \setminus \{X_0\}$. The term $\tilde{P}(x_0 | \mathbf{x}_E, x_j, \pi_j)$ in Eq. (20) is one if π_j is consistent with x_0 and zero otherwise. The sum in the numerator is therefore restricted to (X_j, Π'_j) only.
- (iii) The free variable is observed (i.e., $X_j \in \mathbf{X}_E$). $\tilde{P}(\mathbf{x}_E | x_j, \pi_j)$ in Eq. (20) is one if x_j is consistent with \mathbf{x}_E and zero otherwise. So, in both the numerator and the denominator the sums over x_j can be removed and only the term corresponding to the observed state kept. We similarly proceed if some of the parents of X_j are observed. Thus, in general, we just have to restrict the sums to the variables of (X_j, Π_j) which are not in \mathbf{X}_E .

Coping with zero probabilities. A further discussion is required in the case of zero probabilities. Let us consider a particular iteration of the A-LP algorithm, and note that the coefficients of the objective function in the linear-fractional problem require the computation of conditional probabilities in the Bayesian network with arbitrary specification of the local models of the free variable and the given specification for the others. Yet, this is not possible if the Bayesian network assigns zero probability to a conditioning event. Let us analyze these situations.

²The same procedure can be applied even if constraints among the different credal sets of the same variable are specified. This corresponds to the so-called extensive specification of the credal sets in a credal network (Antonucci and Zaffalon, 2008).

- (i) If there is a $(x_j, \pi_j) \in \Omega_{X_j} \times \Omega_{\Pi_j}$ such that $\tilde{P}(x_j, \pi_j) = 0$, $\tilde{P}(\mathbf{x}_E|x_j, \pi_j)$ which appears in both the numerator and the denominator of Eq. (20) cannot be computed. Yet, in both cases it is multiplied by $\tilde{P}(x_j|\pi_j) \cdot \tilde{P}(\pi_j) = \tilde{P}(x_j, \pi_j)$, which is zero. We can therefore cope with this case by setting the coefficient associated with the optimization variable $P(x_j|\pi_j)$ equal to zero.
- (ii) If there is a $(x_j, \pi_j) \in \Omega_{X_j} \times \Omega_{\Pi_j}$ such that $\tilde{P}(x_j, \pi_j, \mathbf{x}_E) = 0$, we proceed in a similar way. The conditional $\tilde{P}(x_0|x_j, \pi_j, \mathbf{x}_E)$ is multiplied by a null term and the coefficient associated with $P(x_j|\pi_j)$ can be therefore safely set equal to zero.

Finally consider the zero-probability issues for the conditioning event $\mathbf{X}_E = \mathbf{x}_E$. Remember that we are discussing a single iteration of the algorithm. First solve the two linear programs leading to $\underline{P}'(\mathbf{x}_E)$ and $\overline{P}'(\mathbf{x}_E)$. Joint queries like these can be reduced to single-variable queries by the technique proposed by Mauá et al. (2012b).

If $\overline{P}'(\mathbf{x}_E) = 0$, all the models corresponding to the current choice of the extreme points of the non-free variables assign zero probability to the conditioning event. Another free variable, possibly leading to strictly positive upper probability, should be picked instead. If this is not possible, we adopt the natural extension of the model (Walley, 1991), which corresponds to set $\underline{P}'(x_0|\mathbf{x}_E) = 0$ and $\overline{P}'(x_0|\mathbf{x}_E) = 1$.

If $\overline{P}'(\mathbf{x}_E) > 0$ but $\underline{P}'(\mathbf{x}_E) = 0$, there is at least a specification of the local models of the free variable that assigns zero probability to the conditioning event. If this is the case, we adopt the regular extension (Walley, 1991, App. J), which discards specifications of this kind. The constraint $P(\mathbf{x}_E) > 0$ is therefore added to Eq. (20) by requiring the strict positivity of the denominator of the objective function. As linear solvers cannot cope with strict inequalities, the constraint is converted into a non-strict one with the machine epsilon instead of the zero.

4.4. Computational complexity of A-LP

Let m and l denote, respectively, the maximum number of states and incoming parents (i.e., the indegree) of the network variables: $m := \max_{i=0,\dots,n} |\Omega_{X_i}|$ and $l := \max_{i=0,\dots,n} |\Pi_i|$. Let q be the maximum number of linear constraints required to define a local credal set. A linear program as in Eq. (11) has at most m^{l+1} variables and $m^l \cdot q$ constraints. Because the size of the input should be already proportional to $m^l \cdot q$, the algorithm spends time equivalent to run a linear programming solver on the (local) input specification times the total number of iterations. Moreover, to compute the coefficients of the objective function in Eq. (20) a constant number of Bayesian inferences is required for each optimization variable, this task being exponential in the treewidth of \mathcal{G} .

In our algorithm the local credal sets are assumed to be defined by linear constraints. This is a major difference with the algorithm of da Rocha et al. (2003), which requires instead the availability of the extreme points of the local models. Polyhedra algorithms (e.g., Avis, 2000) can be used to compute the extreme points of a credal set defined by linear constraints and vice versa. Yet, both these conversions can induce an exponential growth (Wallner, 2007; Avis, 2000). The choice between the two algorithms should take this issue into account.

4.5. Coping with other independence concepts

The A-LP algorithm is designed to compute approximate inferences based on the strong extension of a credal network. Such an extension, defined as in Eq. (10), is based on the notion of strong independence for credal sets. The notion of *epistemic* irrelevance is an alternative independence concept which has been also considered for credal networks (e.g., de Bock and de Cooman, 2014). The corresponding extension, called epistemic, is a larger credal set possibly giving more conservative inferences. The inferences provided by A-LP can be therefore regarded as an inner approximation also for inferences based on the epistemic extension. Yet, there is evidence in the literature that even an exact computation based on the strong extension can be an inaccurate inner approximation for the epistemic extension (de Cooman et al., 2010). We therefore do not recommend to use A-LP in the epistemic case, apart from the special situations, considered by Mauá et al. (2014), in which the inferences associated with the two extensions coincide. A similar situation holds for the (less known) Kuznetsov independence, as it is also a larger extension than the strong and results can significantly differ (Cozman and de Campos, 2014).

5. Decision making with credal networks

Like Bayesian networks, credal networks are often used to implement classifiers and knowledge-based decision-support systems (Corani et al., 2014, 2012; Antonucci et al., 2014). In these applications the goal is not only to compute the posterior probabilities but also to determine the most likely state for the variable of interest X_0 given the observation \mathbf{x}_E . This can be based on the decision criteria discussed in the second part of Sect. 3.2.

For Bayesian networks the problem simply reduces to a number of updating tasks, as the most probable a posteriori state is $x_0^* := \arg \max_{x_0 \in \Omega_{X_0}} P(x_0 | \mathbf{x}_E)$. Decisions based on the Γ -maximin/max and interval dominance criteria can be similarly computed in credal networks. This can be achieved by considering the posterior intervals only, i.e., $\{\underline{P}(x_0 | \mathbf{x}_E), \bar{P}(x_0 | \mathbf{x}_E)\}_{x_0 \in \Omega_{X_0}}$. The A-LP algorithm can be therefore used to directly address decision making based on those criteria. Maximality and E-admissibility require instead, at least in their definitions, the availability of the credal set $K(X_0 | \mathbf{x}_E)$ in this case). In this part of the paper we fill this gap by reducing decision making based on maximality to a number of updating tasks (Sect. 5.2), and by developing a variant of the A-LP algorithm able to detect the E-admissible options (Sect. 5.3). To derive these results, we first reduce to a single updating task the computation of the lower and upper expectations of a gamble over a variable in a credal network. This reduction has been originally presented by Antonucci and de Campos (2011).

5.1. Computing expectations

Consider the lower expectation $\underline{E}(g)$ in Eq. (4). Regard the credal set $K(X)$ as the local model, but also the strong extension, of a credal network with a single variable. Augment the network with a variable Y , with $\Omega_Y := \{0, 1\}$, which is assumed to be a child of X . As local models for Y given X , adopt a ‘Bayesian’ specification: each

conditional credal set is made of a single element. Assessing the values of $P(Y = 0|x)$ for each $x \in \Omega_X$ is therefore a sufficient specification. The lower probability for the first state of the auxiliary variable with respect to the augmented credal network is:

$$\underline{P}(Y = 0) = \min_{P(X) \in K(X)} \sum_{x \in \Omega_X} P(x) \cdot P(Y = 0|x). \quad (21)$$

Setting $P(Y = 0|x) := g(x)$ for each $x \in \Omega_X$, Eq. (21) coincides with Eq. (4), i.e., $\underline{E}(g) = \underline{P}(Y = 0)$. In other words, an appropriate quantification of the conditional probabilities for the auxiliary variable makes the lower (and similarly the upper) expectation of a gamble equal to the lower (upper) probability of the auxiliary variable.

In the above derivation the gamble g should have values in the $[0, 1]$ interval. If this is not the case, we can normalize its values as follows:

$$\tilde{g}(x) := \frac{g(x) - \min_{x \in \Omega_X} g(x)}{\max_{x \in \Omega_X} g(x) - \min_{x \in \Omega_X} g(x)}. \quad (22)$$

The values of gamble \tilde{g} can be regarded as probabilities and its lower expectation computed as in Eq. (21). Thus, for the original gamble:

$$\underline{E}(g) = \min_{x \in \Omega_X} g(x) + [\max_{x \in \Omega_X} g(x) - \min_{x \in \Omega_X} g(x)] \cdot \underline{E}(\tilde{g}). \quad (23)$$

This is based on the fact that, given a gamble f , if α and β are real constants, with α positive, $\underline{E}(\alpha f + \beta) = \alpha \underline{E}(f) + \beta$ (Walley, 1991).

The augmentation of Y to X is a local transformation which can be equivalently performed in networks with more than a single variable, by simply replacing $K(X)$ with $K(\mathbf{X})$ as in Eq. (10). Similarly, the derivation holds even if the set of observed variables \mathbf{X}_E is not empty. Summarising, in a generic credal network over \mathbf{X} , we can express the conditional lower expectation of a (normalised) gamble over a queried variable X_0 given evidence \mathbf{x}_E as a posterior lower probability in an augmented network, i.e., $\underline{E}(\tilde{g}|\mathbf{x}_E) = \underline{P}(Y = 0|\mathbf{x}_E)$. Thus, A-LP can be also used to evaluate expectations.

5.2. Maximality-based decision making

To practically compute the maximal states of the queried variable X_0 in a credal network given the observation \mathbf{x}_E we proceed as follows. The set of optimal options $\Omega_{X_0}^*$ is initialized to Ω_{X_0} . Then, for each $x'_0, x''_0 \in \Omega_{X_0}$, the following relation should be checked:

$$\min_{P(X_0|\mathbf{x}_E) \in K(X_0|\mathbf{x}_E)} [P(x''_0|\mathbf{x}_E) - P(x'_0|\mathbf{x}_E)] > 0, \quad (24)$$

and, every time it is satisfied, x'_0 is removed from $\Omega_{X_0}^*$.

Complexity analysis. Let us call *maximality dominance test* the evaluation in Eq. (24). Remember that the treewidth of a network measures the extent to which it resembles a tree (Koller and N. Friedman, 2009), and that we assume all the states are mentioned in the input specification. The computational complexity of maximality-based decision making in credal networks can be characterized by the following result.

Theorem 1. *The maximality dominance test is coNP-complete in bounded treewidth networks and coNP^{PP}-complete in networks of general topology, irrespective of the number of states in the queried variable.*

Hence, deciding whether a class is in the maximal set is a very demanding query, because it is tested against all other classes, and each of such tests can be itself hard. Nevertheless, the problem of deciding whether a state x'_0 is maximal can be shown to be in NP (respectively NP^{PP} for general topology).

Theorem 2. *Deciding whether a given state of a queried variable is maximal is NP-complete in bounded treewidth networks and NP^{PP}-complete in networks of general topology, irrespective of the number of states in the queried variable.*

Solving Maximality. The maximality dominance test in Eq. (24) can be checked by evaluating, for each $x'_0, x''_0 \in \Omega_{X_0}$, the lower expectation of the gamble:

$$g_{x'_0, x''_0}(x_0) = \begin{cases} -1 & \text{if } x_0 = x'_0 \\ +1 & \text{if } x_0 = x''_0 \\ 0 & \text{otherwise.} \end{cases} \quad (25)$$

The task can be reduced to an updating task by using the transformation described in the previous section. This means to augment X_0 with an auxiliary variable Y such that:

$$P(Y = 0|x_0) = \begin{cases} 0 & \text{if } x_0 = x'_0 \\ 1 & \text{if } x_0 = x''_0 \\ \frac{1}{2} & \text{otherwise.} \end{cases} \quad (26)$$

The dominance test in Eq. (24) is equivalent to check whether $\underline{P}(Y = 0|x_E) > \frac{1}{2}$. The A-LP algorithm can be therefore used to address the dominance test for each pair of classes and determine the undominated ones according to maximality. As A-LP returns an upper approximation $\underline{P}'(Y = 0|x_E) \geq \underline{P}(Y = 0|x_E)$ some dominance detected by the algorithm might not really take place. Hence, the set of optimal classes evaluated by the approximate algorithm might be a subset of the exact one.

5.3. Decisions based on E-admissibility

Following the definition of E-admissibility in the second part of Sect. 3.1, the condition for $x_0 \in \Omega_{X_0}$ being an admissible option in a credal network with queried variable X_0 , strong extension $K(\mathbf{X})$, and observation \mathbf{x}_E is:

$$\exists P(\mathbf{X}) \in K(\mathbf{X}) : \forall x'_0 \in \Omega_{X_0} : P(x_0|\mathbf{x}_E) \geq P(x'_0|\mathbf{x}_E). \quad (27)$$

Complexity analysis. Let us call *E-admissibility test* the evaluation in Eq. (27). The computational complexity of decision making based on E-admissibility can be characterized exactly like that based on maximality by the following result.

Theorem 3. *The E-admissibility test is NP-complete in bounded treewidth networks and NP^{PP}-complete in networks of general topology, irrespective of the number of states in the queried variable.*

Solving E-admissibility. The E-admissibility decision test for $x_0 \in \Omega_{X_0}$ in Eq. (27) is equivalent to check whether or not the following constraints can be all satisfied:

$$\exists P(\mathbf{X}) \in K(\mathbf{X}) : \forall x'_0 \in \Omega_{X_0} : P(x_0, \mathbf{x}_E) \geq P(x'_0, \mathbf{x}_E), \quad (28)$$

as long as only specifications assigning strictly positive probability to the evidence are considered, this being consistent with the regular extension. We are not aware of any local transformation to the network that can be used to solve the E-admissibility decision test, and unfortunately the attempt appearing in Antonucci and de Campos (2011) is invalid. In spite of that, we can reduce the E-admissibility decision test to the problem of whether the following optimization problem has a zero optimal value.

$$\begin{aligned} & \text{minimize} && w, \\ & \text{subject to} && 0 \leq w \leq 1, \\ & && P(X_i | \pi_i) \in K(X_i | \pi_i), \quad \forall i, \forall \pi_i, \\ & && P(x_0, \mathbf{x}_E) \geq P(x'_0, \mathbf{x}_E) - w, \quad \forall x'_0 \in \Omega_{X_0}. \end{aligned}$$

This problem has always a feasible solution by choosing $w = 1$, while a feasible solution with $w = 0$ exists if and only if x_0 is E-admissible. The global constraint about the joint probability belonging to the strong extension has been replaced by local ones; this is possible as the convexification in Eq. (10) does not affect the extreme points. The optimization variables are the local probabilities $P(x_i | \pi_i)$ for each $x_i \in \Omega_{X_i}$, $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \dots, n$, and the slack variable w . Apart from those in the last line, all the other constraints are linear. The problem rewrites as follows.

$$\begin{aligned} & \text{minimize} && w, \\ & \text{subject to} && 0 \leq w \leq 1, \\ & && P(X_i | \pi_i) \in K(X_i | \pi_i), \quad \forall i, \forall \pi_i, \\ & && \sum_{x_j, \pi_j} [P(x_0, \mathbf{x}_E | x_j \pi_j) - P(x'_0, \mathbf{x}_E | x_j, \pi_j)] P(\pi_j) P(x_j | \pi_j) + w \geq 0, \quad \forall x'_0 \in \Omega_{X_0}. \end{aligned}$$

Now the constraints in the last line have a clear multilinear form, analogous to Eq. (11). As in the case of A-LP, we add further constraints requiring the local credal sets of the non-free variables to be shrunk to single extreme points $\tilde{P}(X_i | \pi_i) \in \text{ext}[K(X_i | \pi_i)]$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \dots, n, i \neq j$. This produces the following linear problem.

$$\begin{aligned} & \text{minimize} && w, \\ & \text{subject to} && 0 \leq w \leq 1, \\ & && P(X_j | \pi_j) \in K(X_j | \pi_j), \quad \forall \pi_j, \\ & && \sum_{x_j, \pi_j} \left[\tilde{P}(x_0, \mathbf{x}_E | x_j \pi_j) - \tilde{P}(x'_0, \mathbf{x}_E | x_j, \pi_j) \right] \tilde{P}(\pi_j) P(x_j | \pi_j) + w \geq 0, \quad \forall x'_0 \in \Omega_{X_0}. \end{aligned}$$

The optimization variables are now $\{P(x_j | \pi_j)\}_{x_j \in \Omega_{X_j}}^{\pi_j \in \Omega_{\Pi_j}}$ and w . Their coefficients in the constraints in the last line can be computed by inferences in a Bayesian network

with arbitrary specification of the local models for the free variable X_j and the given specification of the other local models. The procedure can be iterated by varying the free variable like in the A-LP algorithm. At any moment that a solution yielding value zero is reached, we may stop and declare that x_0 is E-admissible. Unfortunately an approximate method (as ours) may not be able to find a feasible solution with optimal value zero even if it exists (that happens when the method gets stuck in local optima, for example), and in such case we would not recognize x_0 as admissible. Hence, the proposed algorithm (when it fails to find the correct solution) produces an inner approximation (i.e., a subset) of the set of E-admissible states. This relates to the NP-hardness of the problem.

6. Experiments

In this section we report the results of a numerical validation of the algorithm, by giving also some details about the implementation of the A-LP algorithm.

Benchmark. To validate the accuracy of the inferences computed by A-LP, we use a benchmark made of different credal networks with random topology, either multiply or singly connected, and two classical (multiply connected) models, namely the classical Alarm and Insurance networks. The maximum indegree for the networks with random topology is limited to 5. The number of states for the Alarm and the Insurance networks is the same as in their original specifications, while for the other networks the number of states is randomly chosen between 2 and 8. All the models are quantified by randomly generated conditional credal sets with a fixed number of extreme points, whose number is ranging from 2 to 8 for each network. Both marginal and conditional queries are considered in these models. In the unconditional case, the queried variable is randomly selected. Only variables corresponding to root nodes are considered instead in the conditional case, for which random observations are specified on a number of variables ranging from 1 to 3 and corresponding to leaf nodes. The exact values of these inferences have been computed by mapping the problem to an integer linear program (de Campos and Cozman, 2007), which is solved by CPLEX. The network specifications together with the results of these exact inferences are freely available.³

Software. Inferences are computed by a Java implementation of A-LP, which is available as a free software tool.⁴ The code is implemented on top of the Eclipse platform. The algorithm is packaged into a console application along with a number of other plugins for parsing of specific file formats, model data structures and search strategies. The COIN-OR library for linear programming, accessible to Java with a selection of JNI wrappers, is used to solve the linear tasks.⁵ The Bayesian network inferences are computed with the JTree algorithm of the C++ LibDAI library for inference in graphical models (Mooij, 2010).

³See CNBenchmark section at <http://ipg.idsia.ch/software>.

⁴See A-LP section at <http://ipg.idsia.ch/software>.

⁵See <http://www.coin-or.org>.

Caching is used to speed up the performance. For instance, during the search, the linear problems for each free variable are cached and only the objective function redefined. Since the search does not affect the topology of the network, even the junction tree is cached, and only the conditional probability tables of the free variable are updated.

Results. Credal networks updating is an NP-hard task: the inferences computed with A-LP are therefore compared with other approximate algorithms. We consider the *iterated local search* (ILS, da Rocha et al., 2003) and the *generalized loopy 2U* (GL2U, Antonucci et al., 2010).

Networks	# of tests	A-LP	ILS	GL2U
Alarm	973	.0474	.2709	.1218
Insurance	650	.0767	.2700	.1818
Random (single)	6162	.0816	.1528	.1724
Random (multi)	2963	.0855	.1269	.1594

Table 1: Simulations for unconditional queries

Networks	# of tests	A-LP	ILS	GL2U
Random (single)	97	.0347	.1462	.2359
Random (multi)	208	.0227	.1215	.2979

Table 2: Simulations for conditional queries

Before commenting on the results, note that our approach assumes the local credal sets to be specified by linear constraints. This is often the case in real scenarios (e.g., credal classifiers or knowledge-based expert systems quantified by probability intervals). Conversely, the credal networks used for benchmarking represent their local credal sets by explicit enumeration of the extreme points. The reason is that the other algorithms require the local credal sets to be described by their extreme points. To compute inferences with A-LP in the benchmark networks, we first compute the constraint-based representation of the local credal sets (e.g., see Avis, 2000).

Tables 1 and 2 contain the results for, respectively, unconditional and conditional queries. The third column of both tables reports the mean square difference between the inner approximation obtained with A-LP and the exact inferences. The value $s = 10$ for the maximum number of no-improve iterations is used. The number t of random restarts is not fixed, and a one-minute timeout is used instead. The same timeout was used for the other algorithms. The comparison with ILS and GL2U (fourth and fifth columns) clearly illustrates a better performance of the approach proposed in this paper. Note that we focus here on the updating task, as decision making has been reduced to updating in Sect. 5. Note also that an approximation in the updating not necessarily affects the identification of the optimal options.

7. Conclusions and outlooks

A new algorithm based on a sequence of linear optimizations is proposed for approximate credal network updating. The algorithm can deal with a constraint-based specification of credal networks, and provides inner approximate solutions. It is also extended to find the maximal and the E-admissible classes in classification and decision-making problem. The complexities of these problems and of the algorithm are presented. From a practical perspective, empirical results are promising: the algorithm is fast and accurate. As future work, we intend to test the algorithm on larger networks and with other search heuristics. Also the combination of the proposed inner approximation with outer approximations (e.g., Cano et al., 2007) can be considered as a possible direction to obtain reliable inferences.

Appendix A. Proofs

Proof of Prop. 1. *It suffices to put in evidence the terms $\{\tilde{P}(x_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_i}}$ in the definition of $\tilde{P}(x_0)$ and note that, by definition of consistency between Bayesian and credal networks, $\tilde{P}(X_j|\pi_j) \in K(X_j|\pi_j)$ for each $\pi_j \in \Omega_{\Pi_j}$.* \square

Proof of Th. 1. *Given a credal network, evidence \mathbf{x}_E , $q \in \Omega_Q$, and a rational $r \in [0, 1]$, the inference query decides whether there exists $P \in K(\mathbf{X})$ such that $P(q|\mathbf{x}_E) \geq r$ (de Campos and Cozman, 2005).*

We show hardness by demonstrating that the complementary decision, that is, whether the minimization of Eq. (24) is less than or equal to zero, is NP^{PP} -hard in general, and NP -hard for bounded treewidth networks. For that, we reduce the marginal inference problem in a credal network to it. Marginal inference in credal networks is shown to be NP -hard in polytrees with at most two parents per node and NP^{PP} -hard in general networks de Campos and Cozman (2005).

Take a credal network with inference query $\exists P : P(q|\mathbf{x}_E) \geq r$, for a given rational r , query q and evidence \mathbf{x}_E . Build a new network by adding a binary node X_0 , which has Q as sole parent and precise probability mass functions defined as $P(x_0''|q) = \frac{r}{2}$ and $P(x_0''|\neg q) = \frac{1+r}{2}$. Note that the new network has the same topology (and treewidth) of the original one. Now, the complement of the dominance test asks whether

$$\min_P [P(x_0''|\mathbf{x}_E) - P(x_0'|\mathbf{x}_E)] \leq 0 \iff \min_P [2P(x_0''|\mathbf{x}_E) - 1] \leq 0$$

$$\iff \min_P [rP(q|\mathbf{x}_E) + (1+r)P(\neg q|\mathbf{x}_E)) - 1] \leq 0 \iff$$

$$\min_P [r - P(q|\mathbf{x}_E)] \leq 0 \iff \max_P P(q|\mathbf{x}_E) \geq r \iff \exists P : P(q|\mathbf{x}_E) \geq r,$$

which is exactly the credal network marginal query. As the treewidth of the network has not been modified, the hardness results follow. Pertinence of this complementary decision in NP for the case of bounded treewidth (respectively in NP^{PP} for the general case) is immediate, since given $P \in K(\mathbf{X})$, we can use a Bayesian network inference to certify that $P(x_0''|\mathbf{x}_E) \leq P(x_0'|\mathbf{x}_E)$ (in polynomial time for bounded treewidth nets and by using the PP oracle for the general case). \square

Proof of Th. 2. Let $x'_0 \in \Omega_{X_0}$ be the state of interest of the queried variable X_0 . By negating Eq. (24), we have that x'_0 is maximal if and only if there are mass functions $\{P_{x''_0}(X_0|\mathbf{x}_E)\}_{x''_0 \in \Omega_{X_0}}$ with $P_{x''_0}(X_0|\mathbf{x}_E) \in K(X_0|\mathbf{x}_E)$ for each $x''_0 \in \Omega_{X_0}$, such that, for each $x''_0 \in \Omega_{X_0}$:

$$P_{x''_0}(x'_0|\mathbf{x}_E) \geq P_{x''_0}(x''_0|\mathbf{x}_E). \quad (\text{A.1})$$

Because given $\{P_{x''_0}(X_0|\mathbf{x}_E)\}_{x''_0 \in \Omega_{X_0}}$ we can check in polynomial time whether all these constraints of Expression (A.1) are satisfied (if the network has bounded treewidth, otherwise we need PP machines), the problem is in NP (respectively in NP^{PP} for general topology). Hardness in those classes comes from Th. 1, applied to a case where the queried variable is binary. In that case, if we can decide whether x'_0 is maximal, then we can decide whether

$$\min_{P(X_0|\mathbf{x}_E) \in K(X_0|\mathbf{x}_E)} [P(x''_0|\mathbf{x}_E) - P(x'_0|\mathbf{x}_E)] \leq 0, \text{ with } x''_0 \neq x'_0,$$

which is an NP-hard (respectively NP^{PP} -hard, in general topology) problem, as it is the complement of Expression (24) of Th. 1. \square

Proof of Th. 3. Pertinence is straightfoward from Expr. (27), as given $P(\mathbf{X})$, one can check the satisfiability of the constraints by using $O(|\Omega_{X_0}|)$ queries in the corresponding Bayesian networks (spending polynomial time for bounded treewidth networks, and hence the problem is in NP, while using PP machines for general topology, and in this case the problem is in NP^{PP}). Hardness comes from the application of Th. 2 when the queried variable is binary. Because the E-admissibility set and the maximal set are identical in this case, the hardness follows. \square

Acknowledgement

This work was partly supported by the Swiss NSF grants nos. 200021_146606 / 1 and 200020_137680 / 1. The work of C. P. de Campos has been mostly performed while he was with IDSIA.

References

- A. Antonucci, C. de Campos, D. Huber, M. Zaffalon, Approximating credal network inferences by linear programming, in: L. C. van der Gaag (Ed.), Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, vol. 7958 of *Lecture Notes in Artificial Intelligence*, Springer, Berlin Heidelberg, 13–25, 2013.
- A. Antonucci, C. de Campos, Decision making by credal nets, in: Proceedings of the International Conference on Intelligent Human-machine Systems and Cybernetics (IHMSC 2011), vol. 1, IEEE, Hangzhou (China), 201–204, 2011.
- F. G. Cozman, Credal networks, *Artificial Intelligence* 120 (2000) 199–233.

- J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, 1988.
- G. Shafer, A mathematical theory of evidence, Princeton University Press, 1976.
- D. Dubois, H. Prade, Possibility theory: an approach to computerized processing of uncertainty, Plenum Press, New York, 1988.
- C. P. de Campos, F. G. Cozman, Inference in credal networks using multilinear programming, in: Proceedings of the Second Starting AI Researcher Symposium, IOS Press, Amsterdam, 50–61, 2004.
- C. P. de Campos, F. G. Cozman, The inferential complexity of Bayesian and credal networks, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, 1313–1318, 2005.
- D. Koller, N. N. Friedman, Probabilistic graphical models: principles and techniques, MIT Press, 2009.
- J. C. F. da Rocha, F. G. Cozman, C. P. de Campos, Inference in polytrees with sets of probabilities, in: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, 217–224, 2003.
- A. Cano, M. Gomez, S. Moral, J. Abellán, Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks, International Journal of Approximate Reasoning 44 (3) (2007) 261–280.
- C. P. de Campos, F. G. Cozman, Inference in credal networks through integer programming, in: Proceedings of the Fifth International Symposium on Imprecise Probability: Theories and Applications, Prague, 145–154, 2007.
- A. Antonucci, S. Yi, C. de Campos, M. Zaffalon, Generalized loopy 2U: a new algorithm for approximate inference in credal networks, International Journal of Approximate Reasoning 51 (5) (2010) 474–484.
- D. Mauá, C. de Campos, M. Zaffalon, Updating credal networks is approximable in polynomial time, International Journal of Approximate Reasoning 53 (8) (2012a) 1183–1199.
- G. de Cooman, F. Hermans, A. Antonucci, M. Zaffalon, Epistemic irrelevance in credal nets: the case of imprecise Markov trees, International Journal of Approximate Reasoning 51 (9) (2010) 1029–1052.
- P. Walley, Statistical reasoning with imprecise probabilities, Chapman and Hall, 1991.
- I. Levi, The Enterprise of Knowledge, MIT Press, London, 1980.
- C. P. de Campos, Q. Ji, Improving Bayesian Network parameter learning using constraints, in: 19th International Conference on Pattern Recognition (ICPR), 1–4, doi: 10.1109/ICPR.2008.4761287, 2008.

- M. C. M. Troffaes, Decision making under uncertainty using imprecise probabilities, *International Journal of Approximate Reasoning* 45 (1) (2007) 17–29.
- A. Antonucci, M. Zaffalon, Decision-theoretic specification of credal networks: a unified language for uncertain modeling with sets of Bayesian networks, *International Journal of Approximate Reasoning* 49 (2) (2008) 345–361.
- A. M. Lukatskii, D. V. Shapot, Problems in multilinear programming, *Computational Mathematics and Mathematical Physics* 41(5) (2000) 638–648.
- F. Glover, G. Kochenberger, *Handbook of Metaheuristics*, International series in operations research & management science, Kluwer Academic Publishers, 2003.
- D. Mauá, C. de Campos, A. Benavoli, A. Antonucci, Probabilistic inference in credal networks: new complexity results, *Journal of Artificial Intelligence Research* 50 (2014) 603–637.
- A. Charnes, W. Cooper, Programs with linear fractional functions, *Naval Research Logistics Quarterly* 9 (1962) 181–196.
- D. Mauá, C. de Campos, M. Zaffalon, The complexity of approximately solving influence diagrams, in: *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI 2012)*, 604–613, 2012b.
- D. Avis, lrs: a revised implementation of the reverse search vertex enumeration algorithm, *Polytopes - Combinatorics and Computation* (2000) 177–198 DMV Seminar Band 29.
- A. Wallner, Extreme points of coherent probabilities in finite spaces, *International Journal of Approximate Reasoning* 44 (3) (2007) 339–357.
- J. de Bock, G. de Cooman, Credal networks under epistemic irrelevance: the sets of desirable gambles approach, *International Journal of Approximate Reasoning* (to appear).
- F. G. Cozman, C. P. de Campos, Kuznetsov independence for interval-valued expectations and sets of probability distributions: properties and algorithms, *International Journal of Approximate Reasoning* 55 (2) (2014) 666–682.
- G. Corani, J. Abellán, A. Masegosa, S. Moral, M. Zaffalon, Classification, in: T. Augustin, F. Coolen, G. de Cooman, M. Troffaes (Eds.), *Introduction to Imprecise Probabilities*, Wiley, 261–285, 2014.
- G. Corani, A. Antonucci, M. Zaffalon, Bayesian networks with imprecise probabilities: theory and application to classification, in: D. Holmes, L. Jain (Eds.), *Data Mining: Foundations and Intelligent Paradigms*, vol. 23 of *Intelligent Systems Reference Library*, Springer, Berlin / Heidelberg, 49–93, 2012.
- A. Antonucci, C. de Campos, M. Zaffalon, Probabilistic graphical models, in: T. Augustin, F. Coolen, G. de Cooman, M. Troffaes (Eds.), *Introduction to Imprecise Probabilities*, Wiley, 207–229, 2014.

J. M. Mooij, libDAI: a free and open source C++ library for discrete approximate inference in graphical models, *Journal of Machine Learning Research* 11 (2010) 2169–2173.